

d

**CIRCUIT DIVISION ATG PARTIAL CIRCUIT PROCESSING SYSTEM,  
CIRCUIT DIVISION ATG PARTIAL CIRCUIT PROCESSING METHOD,  
AND STORAGE MEDIUM WHERE CIRCUIT DIVISION ATG PARTIAL  
CIRCUIT PROCESSING METHOD IS WRITTEN**

Patent Number: JP11237450  
Publication date: 1999-08-31  
Inventor(s): TOUMIYA TAMAKI  
Applicant(s):: NEC CORP  
Requested Patent: ☐ JP11237450  
Application Number: JP19980037223 19980219  
Priority Number(s):  
IPC Classification: G01R31/3183 ; G06F11/22  
EC Classification:  
Equivalents: JP3050306B2

**Abstract**

**PROBLEM TO BE SOLVED:** To reduce time required for test pattern generation processing.  
**SOLUTION:** A failure detection difficulty circuit judging means 7 for delaying failure information in a partial circuit and outputting it as a list when the number of failures within the partial circuit is larger than a specific value based on the partial circuit and failure information created by a partial circuit creation means 5 is provided. Further, a test pattern is created based on failure information other than that in the partial circuit that is outputted as a delay list from the failure detection difficulty circuit judging means 7 in an ATG means 10, then a test pattern is created based on the failure information in the partial circuit being outputted as the delay list, and failure detection processing is completed when the failure detection reaches a predetermined target failure detection rate.

Data supplied from the **esp@cenet** database - I2

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-237450

(43) 公開日 平成11年(1999) 8月31日

(51) Int.Cl.<sup>6</sup>

G 0 1 R 31/3183

G 0 6 F 11/22

識別記号

3 1 0

F I

G 0 1 R 31/28

G 0 6 F 11/22

Q

3 1 0 C

審査請求 有 請求項の数13 O L (全 17 頁)

(21) 出願番号 特願平10-37223

(22) 出願日 平成10年(1998) 2月19日

(71) 出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(72) 発明者 東宮 玉紀

東京都港区芝五丁目7番1号 日本電気株式会社内

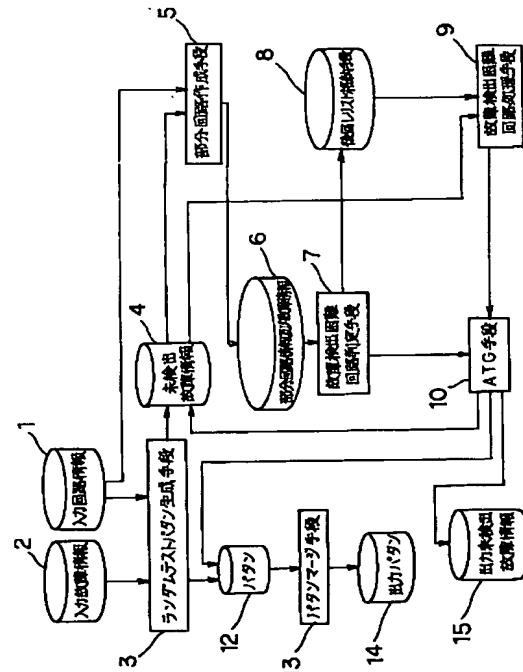
(74) 代理人 弁理士 若林 忠 (外4名)

(54) 【発明の名称】 回路分割ATG部分回路処理方式、回路分割ATG部分回路処理方法及びこれが書き込まれた記憶媒体

(57) 【要約】

【課題】 テストパターン生成処理に費やす時間を短縮する。

【解決手段】 部分回路作成手段5にて作成された部分回路及び故障情報に基づいて、部分回路内の故障の数が所定の値よりも大きな場合、該部分回路における故障情報を後回しリストとして出力する故障検出困難回路判定手段7を設け、さらに、ATG手段10において、故障検出困難回路判定手段7から後回しリストとして出力された部分回路における故障情報以外の故障情報に基づいてテストパターンを生成した後、後回しリストとして出力された部分回路における故障情報に基づいてテストパターンを生成し、故障検出が予め決められた目標故障検出率に達した場合に故障検出処理を終了する。



**【特許請求の範囲】**

【請求項1】 乱数を用いてランダムパターンを発生させるとともに、発生したランダムパターンを用いて回路内の故障を検出し、検出された故障に基づいて、回路内の故障を検出するためのテストパターンを生成するランダムテストパターン生成手段と、回路を分割することにより部分回路を作成するとともに、作成された部分回路にて検出可能な故障情報を生成する部分回路作成手段と、該部分回路作成手段にて作成された部分回路及び故障情報に基づいて、回路内の故障を検出するためのテストパターンを自動生成するATG手段と、前記ランダムテストパターン生成手段及び前記ATG手段にて生成されたテストパターンをマージし、出力パターンとして出力するパターンマージ手段とを有してなる回路分割ATG部分回路処理方式において、

前記部分回路作成手段にて作成された部分回路及び故障情報に基づいて、部分回路内の故障の数が所定の値よりも大きな場合、該部分回路における故障情報を後回しリストとして出力する故障検出困難回路判定手段を有し、前記ATG手段は、前記故障検出困難回路判定手段から後回しリストとして出力された部分回路における故障情報以外の故障情報に基づいて前記テストパターンを生成した後、前記後回しリストとして出力された部分回路における故障情報に基づいて前記テストパターンを生成することを特徴とする回路分割ATG部分回路処理方式。

【請求項2】 請求項1に記載の回路分割ATG部分回路処理方法において、

前記所定の値は、前記ランダムテストパターン生成手段によって検出された故障の数の、回路全体にて定義される故障の数に対する割合に基づいて設定されることを特徴とする回路分割ATG部分回路処理方式。

【請求項3】 請求項1または請求項2に記載の回路分割ATG部分回路処理方式において、前記故障検出困難回路判定手段から後回しリストとして出力された部分回路における故障情報を格納する後回しリスト格納手段を有することを特徴とする回路分割ATG部分回路処理方式。

【請求項4】 請求項3に記載の回路分割ATG部分回路処理方式において、

前記故障検出困難回路判定手段から後回しリストとして出力された部分回路における故障情報以外の故障情報に基づいたテストパターの生成が前記ATG手段にて行われた後、前記後回しリストとして出力された部分回路における故障情報を前記後回しリスト格納手段から抽出し、前記ATG手段に対して出力する故障検出困難回路処理手段を有することを特徴とする回路分割ATG部分回路処理方式。

【請求項5】 請求項4に記載の回路分割ATG部分回路処理方式において、

前記故障検出困難回路処理手段は、前記後回しリスト格

納手段に格納された部分回路のうち、故障数が少ない部分回路における故障情報から順に前記ATG手段に対して出力することを特徴とする回路分割ATG部分回路処理方式。

【請求項6】 請求項1乃至5のいずれか1項に記載の回路分割ATG部分回路処理方法において、

前記ATG手段は、故障検出が予め決められた目標故障検出率に達した場合に故障検出処理を終了することを特徴とする回路分割ATG部分回路処理方式。

【請求項7】 請求項1乃至6のいずれか1項に記載の回路分割ATG部分回路処理方式において、

前記部分回路作成手段は、前記回路を回路内のCPUの数以上の部分回路に分割することを特徴とする回路分割ATG部分回路処理方式。

【請求項8】 乱数を用いて発生したランダムパターンを用いて回路内の故障を検出し、検出された故障に基づいて、回路内の故障を検出するためのテストパターンを生成する第1の処理と、回路を分割することにより部分回路を作成するとともに、作成された部分回路にて検出可能な故障情報を生成する第2の処理と、該第2の処理にて作成された部分回路及び故障情報に基づいて、回路内の故障を検出するためのテストパターンを自動生成する第3の処理と、前記第1及び第3の処理にて生成されたテストパターンをマージし、出力パターンとして出力する第4の処理とを有してなる回路分割ATG部分回路処理方法において、

前記第2の処理にて作成された部分回路及び故障情報に基づいて、部分回路内の故障の数が所定の値よりも大きな場合、該部分回路における故障情報を後回しリストとして出力する第5の処理を有し、

前記第3の処理は、後回しリストとして出力された部分回路における故障情報以外の故障情報に基づいて前記テストパターンを生成した後、前記後回しリストとして出力された部分回路における故障情報に基づいて前記テストパターンを生成することを特徴とする回路分割ATG部分回路処理方法。

【請求項9】 請求項8に記載の回路分割ATG部分回路処理方法において、

前記所定の値は、前記ランダムテストパターン生成手段によって検出された故障の数の、回路全体にて定義される故障の数に対する割合に基づいて設定することを特徴とする回路分割ATG部分回路処理方法。

【請求項10】 請求項8または請求項9に記載の回路分割ATG部分回路処理方法において、前記第3の処理における前記後回しリストとして出力された部分回路の故障情報に基づいた前記テストパターの生成は、前記部分回路内の故障数が少ないものから行うことを特徴とする回路分割ATG部分回路処理方法。

【請求項11】 請求項8乃至10のいずれか1項に記載の回路分割ATG部分回路処理方法において、

前記第3の処理における故障検出が予め決められた目標故障検出率に達した場合に故障検出処理を終了することを特徴とする回路分割ATG部分回路処理方法。

【請求項12】 請求項8乃至11のいずれか1項に記載の回路分割ATG部分回路処理方法において、前記第3の処理における前記後回しリストとして出力された部分回路の故障情報に基づいた前記テストパタンの生成は、前記部分回路における故障を複数のブロックに分割して行うことを特徴とする回路分割ATG部分回路処理方法。

【請求項13】 請求項8乃至12のいずれか1項に記載の回路分割ATG部分回路処理方法が書き込まれたことを特徴とする、コンピュータの処理プログラムが書き込まれた記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、回路を部分回路に分割し、分割された部分回路内の故障パターンに基づいて、故障を検出するためのテストパターンを自動生成する回路分割ATG部分回路処理方式に関する。

【0002】

【従来の技術】ATG (Automatic Test Generation: 自動テスト生成) とは、回路内にて発生する可能性がある故障を定義し、定義された故障を検出するためのテストパターンを、プログラムを用いて自動生成する手法である。

【0003】まず、回路内にて定義された故障を検出するためのランダムパターンを生成し、生成されたランダムパターンを用いて、回路内にて定義された全ての故障に対して故障シミュレーションを行う。

【0004】次に、故障シミュレーションによって検出された故障のパターンをテストパターンとして採用し、検出された故障はそれ以降は故障検出の対象としないようにするランダムテストパターン生成を行う。

【0005】ランダムテストパターン生成にて検出される故障の数が少なくなった後、回路構成を参照しながらテストパターンを自動的に生成するATG処理によってテストパターンを生成する。ここでも、検出された故障はそれ以降は故障検出の対象としないようにする。

【0006】ここで、上述したようなテストパタンの生成においては、目標とする故障検出率（後述）を予め設定し、故障検出率が設定した値に達した場合に処理を終了する方法と、目標とする故障検出率を設定しない方法とがある。

【0007】上述した故障検出率の算出方法においては、例えば以下に示すような方法がある。

【0008】(1) (故障検出率) = (検出された故障の数) / (故障数全体)

(2) (故障検出率) = (検出された故障の数) / { (全故障数) - (検出不可能と判定された故障の数) }

数) }

(3) (故障検出率) = { (検出された故障の数) + (検出不可能と判定された故障の数) } / (全故障数)

ここで、上述したATG処理においては、その処理に莫大な時間がかかる。そこで、処理時間を短縮させるための方法の一つとして分散処理がある。

【0009】回路分割分散処理ATGは、ATG処理を分散して行う方法の一つであり、回路をいくつかの部分回路に分割し、分割された部分回路毎にATGを行うものである。

【0010】部分回路の分割方法は、それぞれの部分回路を独立に処理することが可能となる（処理する順序を任意に入れ替えられる）ような方法であればどのような分割方法でも良い。

【0011】例えば、全ての出力端子をいくつかのグループに分け、1つのグループの出力端子に関係する素子の全てを一つの部分回路とするという方法がある。しかしながらこの方法では、複数の出力端子に影響を与える素子は複数の部分回路に重複して含まれる可能性がある。この場合、複数の部分回路に重複して含まれる素子の入力または出力に定義された故障は、複数の部分回路に重複して含まれることになってしまう。

【0012】他の回路分割の方法でも同様に、複数の部分回路に同じ故障が重複して含まれることがある。部分回路の数は、用いるCPUの数よりも多くなるように分ける。複数のCPUに部分回路のATG処理を割り当て、ATG処理が終了したCPUに次の部分回路のATG処理を割り当てていく。

【0013】図6は、従来の回路分割ATG部分回路処理方式の一構成例を示すブロック図である。

【0014】本従来例は図6に示すように、乱数を用いてランダムパターンを発生させ、発生したランダムパターンとATG処理を行う回路に含まれる全ての素子及びそれらの素子間の接続に関する情報である入力回路情報1とATG処理を行う回路において定義することが可能な故障に関する情報である入力故障情報2とに基づいて、回路内の故障を検出するための故障シミュレーションを行い、該故障シミュレーションによって検出された故障に基づいて、回路内の故障を検出するためのテストパターンを生成し、生成されたテストパターンをボタン12として出力するとともに、故障シミュレーションにおいて検出されなかった故障に関する情報を未検出故障情報4として出力するランダムテストパターン生成手段3と、入力回路情報1に基づいて、ATG処理を行う回路を分割し、部分回路を生成するとともに、生成された部分回路とランダムテストパターン生成手段3から出力された未検出故障情報4とに基づいて、部分回路にて検出可能な故障情報を生成し、部分回路情報及び故障情報6として出力する部分回路作成手段5と、部分回路作成手段5から出力された部分回路情報及び故障情報6に基づいて、回路内

の故障を検出するためのテストボタンを自動生成し、生成したテストボタンをボタン12として出力するとともに、テストボタンの生成に用いられなかった故障情報を未検出故障情報4または出力未検出故障情報15として出力するATG手段10と、ランダムテストボタン生成手段3及びATG手段10にて生成されたテストボタン12をマージし、出力ボタン14として出力するボタンマージ手段13とから構成されている。なお、ATG手段10においては、故障検出率が指定した値に達していない場合にてテストボタンの生成に用いられない故障に関する情報が未検出故障情報4として出力され、故障検出率が指定した値に達した場合にてテストボタンの生成に用いられない故障に関する情報が出力未検出故障情報15として出力される。

【0015】以下に、上記のように構成された回路分割ATG部分回路処理方式におけるテストボタン生成処理について説明する。

【0016】図7は、図6に示した回路分割ATG部分回路処理方式におけるテストボタン生成処理を説明するためのフローチャートである。

【0017】まず、ランダムテストボタン生成手段2において、乱数を用いてランダムボタンが生成され、生成されたランダムボタンと入力故障情報2として格納されている故障のうちまだ検出されていない全ての故障の情報をを用いて故障を検出するための故障シミュレーションが行われ、故障シミュレーションによって検出された故障のボタンがテストボタンとして採用される(ステップS101)。なお、検出された故障はそれ以降のボタン生成の対象から除外される。

【0018】なお、ステップS101における処理においては、部分回路作成前に回路全体に対して行っても良いし、部分回路作成後にそれぞれの部分回路に対して行っても良い。本従来例においては、部分回路作成前に回路全体に対して行っている。また、ランダムテストボタン生成処理においては、検出しやすい故障が検出される傾向があるため、この方法でどれだけの故障を検出できるかがその回路の回路構成が故障を検出しやすいかどうかの指標となる。

【0019】ステップS101において生成されたテストボタンはボタン12として出力され、また、ステップS101における故障シミュレーションにて検出されなかった故障は引き続きテストボタンの生成に用いられるために未検出故障情報4として出力される。

【0020】次に、部分回路作成手段5において、入力回路情報1に基づいて、ATG処理を行う回路が、CPUの数以上に分割され、それにより部分回路が生成されるとともに、生成された部分回路と未検出故障情報4とに基づいて、部分回路にて検出可能な故障情報が生成され、部分回路情報及び故障情報6として出力される(ステップS102)。

【0021】ステップS102にて部分回路作成手段5から出力された部分回路情報及び故障情報6は、回路内のCPUのうち空いているCPUのATG手段10に渡され、ATG手段10において、部分回路情報及び故障情報6に基づいて、回路内の故障を検出するためのテストボタンが自動生成される(ステップS103)。

【0022】ステップS103において生成されたテストボタンはボタン12として出力され、また、ステップS103においてテストボタン生成に用いられなかった故障、すなわち、検出されなかった故障は引き続きテストボタンの生成に用いられるために未検出故障情報4として出力される。

【0023】その後、上述した一連の処理において検出された故障が、予め設定された目標故障検出率を満たすような値となっているかどうか判断され(ステップS104)、目標故障検出率を満たすような値であると判断された場合、ボタンマージ手段13において、ランダムテストボタン生成手段3及びATG手段10にて生成されたテストボタン12がマージされ、出力ボタン14として出力されるとともに、ATG手段10から、テストボタンの生成に用いられない故障に関する情報が出力未検出故障情報15として出力される(ステップS105)。

【0024】一方、ステップS104において、目標故障検出率を満たすような値ではないと判断された場合、全ての部分回路についての処理が終了したかどうか判断され(ステップS106)、全ての部分回路についての処理が終了したと判断された場合、ステップS105における処理に移り、また、全ての部分回路についての処理が終了していないと判断された場合、ステップS102における処理に戻る。

【0025】

【発明が解決しようとする課題】検出困難な故障を検出することは、多大な時間を費やすことになる。例えば、1Mゲートを有する回路において、全体の故障検出に10数時間費やされるのに対して、その中の1つの検出困難な故障を検出するのに数時間もの時間が費やされてしまう場合がある。

【0026】そのため、上述したような従来の回路分割ATG部分回路処理方式においては、回路内において、検出することが困難な故障が局所的に存在する場合、該故障を有する部分回路が早い順番にて処理されると、後に処理される部分回路内に容易に検出できる故障が存在した場合に、容易に検出できる故障を検出するまでの時間が必要以上に費やされてしまうという問題点がある。

【0027】また、検出することが困難な故障を検出しなくても目標故障検出率に到達できるような場合においても、それらの故障を検出するための処理が行われるため、目標故障検出率に到達するために必要以上の時間を費やしてしまうという問題点がある。

【0028】本発明は、上述したような従来の技術が有する問題点を鑑みてなされたものであって、テストパターン生成処理に費やす時間を短縮することができる回路分割ATG部分回路処理方式を提供することを目的とする。

【0029】

【課題を解決しようとする手段】上記目的を達成するために本発明は、乱数を用いてランダムパターンを発生させるとともに、発生したランダムパターンを用いて回路内の故障を検出し、検出された故障に基づいて、回路内の故障を検出するためのテストパターンを生成するランダムテストパターン生成手段と、回路を分割することにより部分回路を作成するとともに、作成された部分回路にて検出可能な故障情報を生成する部分回路作成手段と、該部分回路作成手段にて作成された部分回路及び故障情報に基づいて、回路内の故障を検出するためのテストパターンを自動生成するATG手段と、前記ランダムテストパターン生成手段及び前記ATG手段にて生成されたテストパターンをマージし、出力パターンとして出力するパターンマージ手段とを有してなる回路分割ATG部分回路処理方式において、前記部分回路作成手段にて作成された部分回路及び故障情報に基づいて、部分回路内の故障の数が所定の値よりも大きな場合、該部分回路における故障情報を後回しリストとして出力する故障検出困難回路判定手段を有し、前記ATG手段は、前記故障検出困難回路判定手段から後回しリストとして出力された部分回路における故障情報以外の故障情報に基づいて前記テストパターンを生成した後、前記後回しリストとして出力された部分回路における故障情報に基づいて前記テストパターンを生成することを特徴とする。

【0030】また、前記所定の値は、前記ランダムテストパターン生成手段によって検出された故障の数の、回路全体にて定義される故障の数に対する割合に基づいて設定されることを特徴とする。

【0031】また、前記故障検出困難回路判定手段から後回しリストとして出力された部分回路における故障情報を格納する後回しリスト格納手段を有することを特徴とする。

【0032】また、前記故障検出困難回路判定手段から後回しリストとして出力された部分回路における故障情報以外の故障情報に基づいたテストパターの生成が前記ATG手段にて行われた後、前記後回しリストとして出力された部分回路における故障情報を前記後回しリスト格納手段から抽出し、前記ATG手段に対して出力する故障検出困難回路処理手段を有することを特徴とする。

【0033】また、前記故障検出困難回路処理手段は、前記後回しリスト格納手段に格納された部分回路のうち、故障数が少ない部分回路における故障情報から順に前記ATG手段に対して出力することを特徴とする。

【0034】また、前記ATG手段は、故障検出が予め

決められた目標故障検出率に達した場合に故障検出処理を終了することを特徴とする。

【0035】また、前記部分回路作成手段は、前記回路を回路内のCPUの数以上の部分回路に分割することを特徴とする。

【0036】また、乱数を用いて発生したランダムパターンを用いて回路内の故障を検出し、検出された故障に基づいて、回路内の故障を検出するためのテストパターンを生成する第1の処理と、回路を分割することにより部分回路を作成するとともに、作成された部分回路にて検出可能な故障情報を生成する第2の処理と、該第2の処理にて作成された部分回路及び故障情報に基づいて、回路内の故障を検出するためのテストパターンを自動生成する第3の処理と、前記第1及び第3の処理にて生成されたテストパターンをマージし、出力パターンとして出力する第4の処理とを有してなる回路分割ATG部分回路処理方法において、前記第2の処理にて作成された部分回路及び故障情報に基づいて、部分回路内の故障の数が所定の値よりも大きな場合、該部分回路における故障情報を後回しリストとして出力する第5の処理を有し、前記第3の処理は、後回しリストとして出力された部分回路における故障情報以外の故障情報に基づいて前記テストパターンを生成した後、前記後回しリストとして出力された部分回路における故障情報に基づいて前記テストパターンを生成することを特徴とする。

【0037】また、前記所定の値は、前記ランダムテストパターン生成手段によって検出された故障の数の、回路全体にて定義される故障の数に対する割合に基づいて設定することを特徴とする。

【0038】また、前記第3の処理における前記後回しリストとして出力された部分回路の故障情報に基づいた前記テストパターの生成は、前記部分回路内の故障数が少ないものから行うことを特徴とする。

【0039】また、前記第3の処理における故障検出が予め決められた目標故障検出率に達した場合に故障検出処理を終了することを特徴とする。

【0040】また、前記第3の処理における前記後回しリストとして出力された部分回路の故障情報に基づいた前記テストパターの生成は、前記部分回路における故障を複数のブロックに分割して行うことを特徴とする。

【0041】また、コンピュータの処理プログラムが書き込まれた記憶媒体において、前記回路分割ATG部分回路処理方法が書き込まれたことを特徴とする。

【0042】（作用）上記のように構成された本発明においては、部分回路作成手段にて作成された部分回路及び故障情報に基づいて、部分回路内の故障の数が所定の値よりも大きな場合、該部分回路における故障情報が後回しリストとして出力され、ATG手段において、後回しリストとして出力された部分回路における故障情報以外の故障情報に基づいてテストパターンが生成された後、

後回しリストとして出力された部分回路における故障情報に基づいてテストボタンが生成される。

【0043】このように、故障検出が困難な回路構成を有する部分回路が後回しリストとされ、処理の順番が後回しになる。それにより、故障検出が困難な回路構成を有する部分回路で処理されるはずであった故障が、故障検出が容易な回路構成を有する部分回路にも重複して含まれていた場合、故障検出が容易な回路構成を有する部分回路にて先に処理が行われ、故障が検出される。

【0044】故障検出が容易な回路構成を有する部分回路にて該故障が先に検出されれば、一度検出された故障はそれ以降はテストボタン生成の対象から除外されるため、後回しになった、故障検出が困難な回路構成を有する困難回路では処理されなくなり、回路分割分散ATG全体として、その分の処理時間が短縮されることになる。

【0045】また、目標故障検出率が100%よりも小さな場合においては、故障検出が困難な回路構成を有する部分回路における故障検出処理を全く行う前に目標故障検出率が達成されれば、故障検出が困難な回路構成を有する部分回路における故障検出処理を全く行う必要がなくなることになり、その場合、故障検出が困難な回路構成を有する部分回路における故障検出処理に要する処理時間を削除することができる。例えば、目標故障検出率を95%とした場合、残りの5%に検出困難な故障が含まれていて、故障検出困難な回路構成を有する部分回路における故障検出処理を行う前にその目標故障検出率が達成されれば、故障検出が困難な回路構成を有する部分回路における故障検出処理を全く行う必要がなくなる。

【0046】通常、ランダムテストボタン生成において大半の故障が検出される。一度検出された故障はそれ以降はテストボタン生成の対象から除外されるので、ランダムテストボタン生成終了後には、回路全体の故障のうち大半がテストボタン生成の対象から除外されている。また、ランダムテストボタン生成においては、生成されるボタンがランダムボタンであるが故に、検出しやすい故障が検出される傾向にある。このことを利用し、ランダムテストボタン生成をある程度行った後で、まだ検出されていない故障が多く残る部分回路は、故障を検出しにくい回路構成になっていると判断し、故障検出困難な部分回路とみなして後回しリストの対象にする。

【0047】

【発明の実施の形態】以下に、本発明の実施の形態について図面を参照して説明する。

【0048】(第1の実施の形態)図1は、本発明の回路分割ATG部分回路処理方式の第1の実施の形態を示すブロック図である。

【0049】本形態は図1に示すように、乱数を用いてランダムボタンを発生させ、発生したランダムボタンと

ATG処理を行う回路に含まれる全ての素子及びそれらの素子間の接続に関する情報である入力回路情報1とATG処理を行う回路において定義することが可能な故障に関する情報である入力故障情報2とに基づいて、回路内の故障を検出するための故障シミュレーションを行い、該故障シミュレーションによって検出された故障に基づいて、回路内の故障を検出するためのテストボタンを生成し、生成されたテストボタンをボタン12として出力するとともに、故障シミュレーションにおいて検出されなかった故障に関する情報を未検出故障情報4として出力するランダムテストボタン生成手段3と、入力回路情報1に基づいて、ATG処理を行う回路を分割し、部分回路を生成するとともに、生成された部分回路とランダムテストボタン生成手段3から出力された未検出故障情報4とに基づいて、部分回路にて検出可能な故障情報を生成し、部分回路情報及び故障情報6として出力する部分回路作成手段5と、部分回路作成手段5から出力された部分回路情報及び故障情報6に基づいて、部分回路内の故障の数が所定の値よりも大きな場合あるいは部分回路情報及び故障情報6に依存する値である場合に、その部分回路における故障情報を後回しリストとして出力するとともに、部分回路内の故障の数が所定の値以下である場合あるいは部分回路情報及び故障情報6に依存する値ではない場合に、その部分回路における故障情報をそのまま出力する故障検出困難回路判定手段7と、故障検出困難回路判定手段7から出力された後回しリストを格納する後回しリスト格納手段8と、部分回路作成手段5から出力された部分回路情報及び故障情報6に基づいて、回路内の故障を検出するためのテストボタンを自動生成し、生成したテストボタンをボタン12として出力するとともに、テストボタンの生成に用いられなかった故障情報を未検出故障情報4または出力未検出故障情報15として出力するATG手段10と、後回しリスト格納手段8に格納された後回しリストをATG手段10に出力する故障検出困難回路処理手段9と、ランダムテストボタン生成手段3及びATG手段10にて生成されたテストボタン12をマージし、出力ボタン14として出力するボタンマージ手段13とから構成されている。なお、ATG手段10においては、故障検出率が指定した値に達していない場合にテストボタンの生成に用いられていない故障に関する情報が未検出故障情報4として出力され、故障検出率が指定した値に達した場合にテストボタンの生成に用いられていない故障に関する情報が出力未検出故障情報15として出力される。また、故障検出困難回路処理手段9は、ATG手段10において、故障検出困難回路判定手段7にて後回しリストとして出力されなかった部分回路情報及び故障情報6に基づいてテストボタンが自動生成された後、故障検出率が予め決められた故障検出率に達していない場合、後回しリスト格納手段8に格納された後回しリストをATG手段

10に対して出力する。

【0050】以下に、上記のように構成された回路分割ATG部分回路処理方式におけるテストパタン生成処理について説明する。

【0051】図2は、図1に示した回路分割ATG部分回路処理方式におけるテストパタン生成処理を説明するためのフローチャートである。

【0052】まず、ランダムテストパタン生成手段2において、乱数を用いてランダムパタンが生成され、生成されたランダムパタンと入力故障情報2として格納されている故障のうちまだ検出されていない全ての故障の情報とを用いて故障を検出するための故障シミュレーションが行われ、故障シミュレーションによって検出された故障のパタンがテストパタンとして採用される（ステップS1）。なお、検出された故障はそれ以降のパタン生成の対象から除外される。

【0053】なお、ステップS1における処理においては、部分回路作成前に回路全体に対して行っても良いし、部分回路作成後にそれぞれの部分回路に対して行っても良い。本形態においては、部分回路作成前に回路全体に対して行っている。また、ランダムテストパタン生成処理においては、検出しやすい故障が検出される傾向があるため、この方法でどれだけの故障を検出できるかがその回路の回路構成が故障を検出しやすいかどうかの指標となる。

【0054】ステップS1において生成されたテストパタンはパタン12として出力され、また、ステップS1における故障シミュレーションにて検出されなかった故障は引き続きテストパタンの生成に用いられるために未検出故障情報4として出力される。

【0055】次に、部分回路作成手段5において、入力回路情報1に基づいて、ATG処理を行う回路が、CPUの数以上に分割され、それにより部分回路が生成されるとともに、生成された部分回路と未検出故障情報4とに基づいて、部分回路にて検出可能な故障情報が生成され、部分回路情報及び故障情報6として出力される（ステップS2）。

【0056】次に、故障検出困難回路判定手段7において、部分回路作成手段5から出力された部分回路情報及び故障情報6に基づいて、部分回路内の故障の数が所定の値よりも大きいかどうか判断される（ステップS3）。

【0057】ここで、部分回路内の故障の数が所定の値よりも大きいかどうか判断することによって、その部分回路が故障検出困難な回路であるかどうか判断されるが、一般的に、故障検出困難な回路としては、入力されたデータが複数のゲートに分岐され、出力において再び1つのゲートに集められるようなものが挙げられ、分岐された先でもさらに分岐されるというもの等は、故障検出がさらに困難な回路として挙げられる。なお、故障

検出困難な回路であるかどうかを判断するための値においては、予め決められていてもよいが、ランダムテストパタン生成手段3において検出された故障の数の、回路全体にて定義される故障の数に対する割合に基づいて設定される。

【0058】ステップS3において部分回路内の故障の数が所定の値よりも大きいと判断された場合は、故障検出困難回路判定手段7において、その部分回路における故障情報が後回しリストとして出力され、後回しリスト格納手段8に格納される（ステップS4）。

【0059】一方、ステップS3において部分回路内の故障の数が所定の値以下であると判断された場合は、その部分回路における故障情報が、回路内のCPUのうち空いているCPUのATG手段10に渡され、ATG手段10において、部分回路情報及び故障情報6に基づいて、回路内の故障を検出するためのテストパタンが自動生成される（ステップS5）。

【0060】ステップS5において生成されたテストパタンはパタン12として出力され、また、ステップS5においてテストパタン生成に用いられなかった故障、すなわち、検出されなかった故障は引き続きテストパタンの生成に用いられるために未検出故障情報4として出力される。

【0061】その後、上述した一連の処理において検出された故障の数が、予め設定された目標故障検出率を満たすような値となっているかどうか判断され（ステップS6）、目標故障検出率を満たすような値であると判断された場合、パタンマージ手段13において、ランダムテストパタン生成手段3及びATG手段10にて生成されたテストパタン12がマージされ、出力パタン14として出力されるとともに、ATG手段10から、テストパタンの生成に用いられていない故障に関する情報が出力未検出故障情報15として出力される（ステップS7）。

【0062】一方、ステップS6において、目標故障検出率を満たすような値ではないと判断された場合、全ての部分回路についての処理が終了したかどうか判断され（ステップS8）、全ての部分回路についての処理が終了していないと判断された場合、ステップS2における処理に戻る。

【0063】一方、ステップS8において、全ての部分回路についての処理が終了したと判断された場合、後回しリスト格納手段8内に後回しリストが格納されているかどうか検出され（ステップS9）、後回しリストが1つも格納されていない場合、ステップS7における処理に移る。

【0064】一方、ステップS9において、後回しリスト格納手段8内に後回しリストが格納されていることが検出された場合、故障検出困難回路処理手段9によって、後回しリスト格納手段8に格納された後回しリスト



(部分回路)が、故障数の少ないものから順にソートされる(ステップS10)。これは、ランダムテストボタン生成後に残った故障数が部分回路の回路構成の故障検出しやすさの指標になるためであり、このようにソートすることで、後回しになった部分回路が複数ある場合に、その中で故障検出を最も行いやすい回路構成を有する回路から順に処理することができる。

【0065】次に、故障検出困難回路処理手段9において、未検出故障情報4内の回路全体の未検出故障リストから部分回路に対応する故障が選出され、この故障の数と、回路全体の故障検出率が目標故障検出率に達するために検出される必要のある故障数とが比較され、それにより、部分回路の目標故障検出率(以下、部分回路目標故障検出率と称する)が算出される(ステップS11)。

【0066】次に、ATG手段10において、ステップS11にて算出された部分回路目標故障検出率と、部分回路作成手段5から出力された部分回路情報及び故障情報6とに基づいて、回路内の故障を検出するためのテストボタンが自動生成される(ステップS12)。

【0067】ステップS12において生成されたテストボタンはボタン12として出力され、また、ステップS5においてテストボタン生成に用いられなかった故障、すなわち、検出されなかった故障は、未検出故障情報4として出力される。

【0068】その後、部分回路の故障検出率が、部分回路目標故障検出率を満たしているかどうか判断される(ステップS13)、部分回路目標故障検出率を満たしているを判断された場合、ステップS7における処理に移る。

【0069】一方、ステップS13において、部分回路目標故障検出率を満たしていないと判断された場合、後回しリスト格納手段8に後回しリストが1つも格納されていないかどうか判断される(ステップS14)、後回しリストが1つも格納されていないと判断された場合はステップS7における処理に移り、後回しリストが格納されていると判断された場合はステップS11における処理に戻る。

【0070】このように、ステップS11～S14までの処理が、回路内におけるいずれかのCPUのATG手段10にてテストボタンの自動生成が終了する度に行われ、いずれかの部分回路の故障検出率が部分回路目標故障検出率に達するか、または、後回しリスト9に格納された後回しリスト(部分回路)がなくなるまで行われる。

【0071】(第2の実施の形態)図3は、本発明の回路分割ATG部分回路処理方式の第2の実施の形態を示すブロック図である。

【0072】本形態は図3に示すように、第1の実施の形態において示したものと比べて、部分回路作成手段5

にて部分回路を作成した後、ランダムテストボタン生成手段3にてランダムテストボタンを生成する点のみが異なり、その他の構成については、第1の実施の形態において示したものと同様である。

【0073】本形態に示すようにランダムテストボタン生成を部分回路作成後に行う場合においても、ランダムテストボタン生成後に、ランダムテストボタン生成に用いられなかった故障の数を故障検出困難な部分回路であるかどうかの指標にするという点において、ランダムテストボタン生成を部分回路作成前に行う第1の実施の形態に示したものと同一効果が得られる。

【0074】(第3の実施の形態)以下に、後回しリストの処理を故障分割で行う実施の形態について説明する。

【0075】図4は、本発明の回路分割ATG部分回路処理方法において後回しリストの処理を故障分割で行う実施の形態を説明するためのフローチャートである。なお、故障分割とは、一つの回路で処理する故障を同時に全部与えず、故障をいくつかのグループに分けてグループずつ与える方法である。故障分割で分散処理を行う場合は、複数のCPUに同じ回路を与え、いくつかのグループに分けた故障の1グループずつを違うCPUに与える。

【0076】まず、ATG処理を行う回路が、CPUの数以上に分割され、それにより部分回路が生成されるとともに、部分回路にて検出可能な故障情報が生成される(ステップS201)。

【0077】次に、乱数を用いてランダムボタンが生成され、生成されたランダムボタンと入力故障情報として記述されている故障のうちまだ検出されていない全ての故障の情報とを用いて故障を検出するための故障シミュレーションが行われ、故障シミュレーションによって検出された故障のボタンがテストボタンとして採用される(ステップS202)。なお、検出された故障はそれ以降のボタン生成の対象から除外される。

【0078】次に、部分回路内の故障の数が所定の値よりも大きいかどうか判断される(ステップS203)。

【0079】ここで、部分回路内の故障の数が所定の値よりも大きいかどうかを判断することによって、その部分回路が故障検出困難な回路であるかどうか判断されるが、一般的に、故障検出困難な回路としては、入力されたデータが複数のゲートに分岐され、出力において再び1つのゲートに集められるようなものが挙げられ、分岐された先でもさらに分岐されるというもの等は、故障検出がさらに困難な回路として挙げられる。なお、故障検出困難な回路であるかどうかを判断するための値においては、予め決められていてもよいが、ランダムテストボタン生成手段において検出された故障の数の、回路全体にて定義される故障の数に対する割合に基づいて設定

される。

【0080】ステップS203において部分回路内の故障の数が所定の値よりも大きいと判断された場合は、その部分回路における故障情報が後回しリストとして出力され、後回しリスト格納手段に格納される（ステップS204）。

【0081】一方、ステップS203において部分回路内の故障の数が所定の値以下であると判断された場合は、故障に対するパタンの回路構成に基づいて、回路内の故障を検出するためのテストパターンが自動生成される（ステップS205）。

【0082】その後、上述した一連の処理において検出された故障の数が、予め設定された目標故障検出率を満たすような値となっているかどうか判断され（ステップS206）、目標故障検出率を満たすような値であると判断された場合、ステップS202及びステップS205にて生成されたテストパターンがマージされる（ステップS207）。

【0083】一方、ステップS206において、目標故障検出率を満たすような値ではないと判断された場合、全ての部分回路についての処理が終了したかどうか判断され（ステップS208）、全ての部分回路についての処理が終了していないと判断された場合、ステップS201における処理に戻る。

【0084】一方、ステップS208において、全ての部分回路についての処理が終了したと判断された場合、後回しリスト格納手段内に後回しリストが格納されているかどうか検出され（ステップS209）、後回しリストが1つも格納されていない場合、ステップS207における処理に移る。

【0085】一方、ステップS209において、後回しリスト格納手段内に後回しリストが格納されていることが検出された場合、後回しリスト格納手段に格納された後回しリスト（部分回路）が、故障数の少ないものから順にソートされる（ステップS210）。

【0086】次に、1つの部分回路内の故障がいくつかのグループにランダムに分割され（ステップS211）、回路内の空いているCPUに、分割された故障がグループ毎に渡され、CPUそれぞれにおいてテストパタンの生成が行われる（ステップS212）。

【0087】その後、上述した一連の処理において検出された故障が、予め設定された目標故障検出率を満たすような値となっているかどうか判断され（ステップS213）、目標故障検出率を満たすような値であると判断された場合、ステップS207における処理に移る。

【0088】一方、ステップS213において、目標故障検出率を満たすような値ではないと判断された場合、分割された故障のうち、まだテストパタンの生成に用いられていない故障があるかどうか検出され（ステップS214）、テストパタンの生成に用いられていない故

障があることが検出された場合、ステップS212における処理に戻る。

【0089】一方、ステップS214において、テストパタンの生成に用いられていない故障がないことが検出された場合、後回しリスト格納手段内に後回しリストが格納されているかどうか検出され（ステップS215）、後回しリストが1つも格納されていない場合、ステップS207における処理に移り、後回しリストが格納されている場合、ステップS212における処理に戻る。

【0090】本形態に示したような、後回しリストの処理を故障分割で行うものにおいては、以下に記載するような効果がある。

【0091】回路分割による分散処理においては、同じ故障を複数のCPUで同時に扱うことが起り得るが、検出困難な二つの回路で同時に同じ故障を処理することは、処理時間の無駄になる。

【0092】後回しリストの処理においては、一つの故障を検出するのにかかる時間が後回しにならない部分回路で一つの故障を検出する時間に比べて大きいと考えられるため、このオーバーヘッドが非常に大きい。

【0093】後回しリストを故障分割で処理することによって、全てのCPUで同じ回路を処理し、異なるCPUで同じ故障を扱うことがなくなる。つまり、異なるCPUで同じ故障を処理してしまうことによるオーバーヘッドが避けられる。

【0094】また、後回しリストの処理を故障分割で行うものにおいては、検出困難回路が最後の回路である場合に1つのCPUだけでこれを処理させるよりも効率が良いという効果もある。

【0095】（第4の実施の形態）以下に、上述した実施の形態において示した部分回路の作成方法について具体例を挙げて詳細に説明する。

【0096】図5は、本発明の回路分割ATG部分回路処理方法における部分回路の作成方法の実施の一形態を説明するためのフローチャートである。

【0097】まず、ATG処理を行う回路の情報及び該回路の故障情報が読み込まれる（ステップS301）。

【0098】次に、PO（Primary Output）側から順次入力側のゲートをたどっていき、各POのそれぞれのコーンに含まれるコーンの数がカウントされる（ステップS302）。ここで、コーンとは、PO側から順次入力側のゲートをたどっていきPO（Primary Input）までに得られるゲートの集合のことをいう。

【0099】次に、各POのそれぞれのコーンに含まれるゲートの数に基づいて、ゲートの数が少ないものからPOが順次ソートされ、POリストが作成される（ステップS303）。

【0100】次に、最後にソートされたPOのコーンに含まれるゲートの数が、後に作成される部分回路のゲー

ト数のしきい値として設定される（ステップS304）。

【0101】その後、ランダムテストボタンが生成される（ステップS305）。

【0102】次に、ソートされた順にP OリストからP Oが1つずつ取り出され（ステップS306）、そのP Oのコーンに含まれるゲートのリストが作成される（ステップS307）。

【0103】次に、ステップS307にて作成されたリストに基づいて、ステップS306にて取り出されたP Oのコーンに含まれるゲートの数が部分回路の合計ゲート数に加えられる（ステップS308）。

【0104】次に、ステップS308における、ゲート数の合計が、ステップS304にて設定されたしきい値を越えているかどうか判断され（ステップS309）、しきい値を越えていると判断された場合、その部分回路内の故障リストが作成され（ステップS310）、また、しきい値を越えていないと判断された場合は、ステップS306における処理に戻る。

【0105】次に、ステップS307にて作成されたゲートのリストと、ステップS310にて作成された故障リストがリモートCPUに送られる（ステップS311）。

【0106】その後、リモートCPUにおいて、ステップS311にて送られてきたゲートのリストと故障リストとに基づいてATG処理が行われる（ステップS312）。

【0107】その後の処理においては、上述した実施の形態において説明したものと同様であるため、ここでの説明は省略する。

【0108】なお、リモートCPUにおいてATG処理が終了すると、その旨の信号が部分回路作成手段に送られ、ステップS306における処理に戻る。

【0109】なお、上述した回路分割ATG部分回路処理方法においては、ROM等の、コンピュータの処理プログラムが書き込まれる記憶媒体に書き込まれ、実施する際に該記憶媒体から読み出される。

【0110】

【発明の効果】以上説明したように本発明においては、分回路作成手段にて作成された部分回路及び故障情報に基づいて、部分回路内の故障の数が所定の値よりも大きな場合、該部分回路における故障情報が後回しリストとして出力され、ATG手段において、後回しリストとして出力された部分回路における故障情報以外の故障情報に基づいてテストボタンが生成された後、後回しリストとして出力された部分回路における故障情報に基づいてテストボタンが生成される構成とし、故障検出が予め決められた目標故障検出率に達した場合に故障検出処理を終了するため、故障検出が困難な回路構成を有する部分回路を処理する前に目標故障検出率に達した場合、故障

検出が困難な回路構成を有する部分回路の処理を行う必要がなくなり、回路分割によるATGに要する処理時間を大幅に削減することができる。

【0111】また、故障検出が困難な回路構成を有する部分回路で処理されるはずであった故障が、故障検出が容易な回路構成を有する部分回路にも重複して含まれていた場合、故障検出が容易な回路構成を有する部分回路にて先に処理が行われ、故障が検出され、検出された故障はそれ以降はテストボタン生成の対象から除外されるため、後回しになった故障検出が困難な回路構成を有する困難回路では処理されなくなり、回路分割分散ATG全体として、その分の処理時間を短縮することができる。

【0112】また、後回しリストとされる部分回路にて検出する必要のある故障数が、対応する未検出故障リストの全てであるとは限らないため、この部分回路に必要な故障検出率を計算して指定することで、処理する故障数を減らすことができる。

【図面の簡単な説明】

【図1】本発明の回路分割ATG部分回路処理方式の第1の実施の形態を示すブロック図である。

【図2】図1に示した回路分割ATG部分回路処理方式におけるテストボタン生成処理を説明するためのフローチャートである。

【図3】本発明の回路分割ATG部分回路処理方式の第2の実施の形態を示すブロック図である。

【図4】本発明の回路分割ATG部分回路処理方法において後回しリストの処理を故障分割で行う実施の形態を説明するためのフローチャートである。

【図5】本発明の回路分割ATG部分回路処理方法における部分回路の作成方法の実施の一形態を説明するためのフローチャートである。

【図6】従来の回路分割ATG部分回路処理方式の一構成例を示すブロック図である。

【図7】図6に示した回路分割ATG部分回路処理方式におけるテストボタン生成処理を説明するためのフローチャートである。

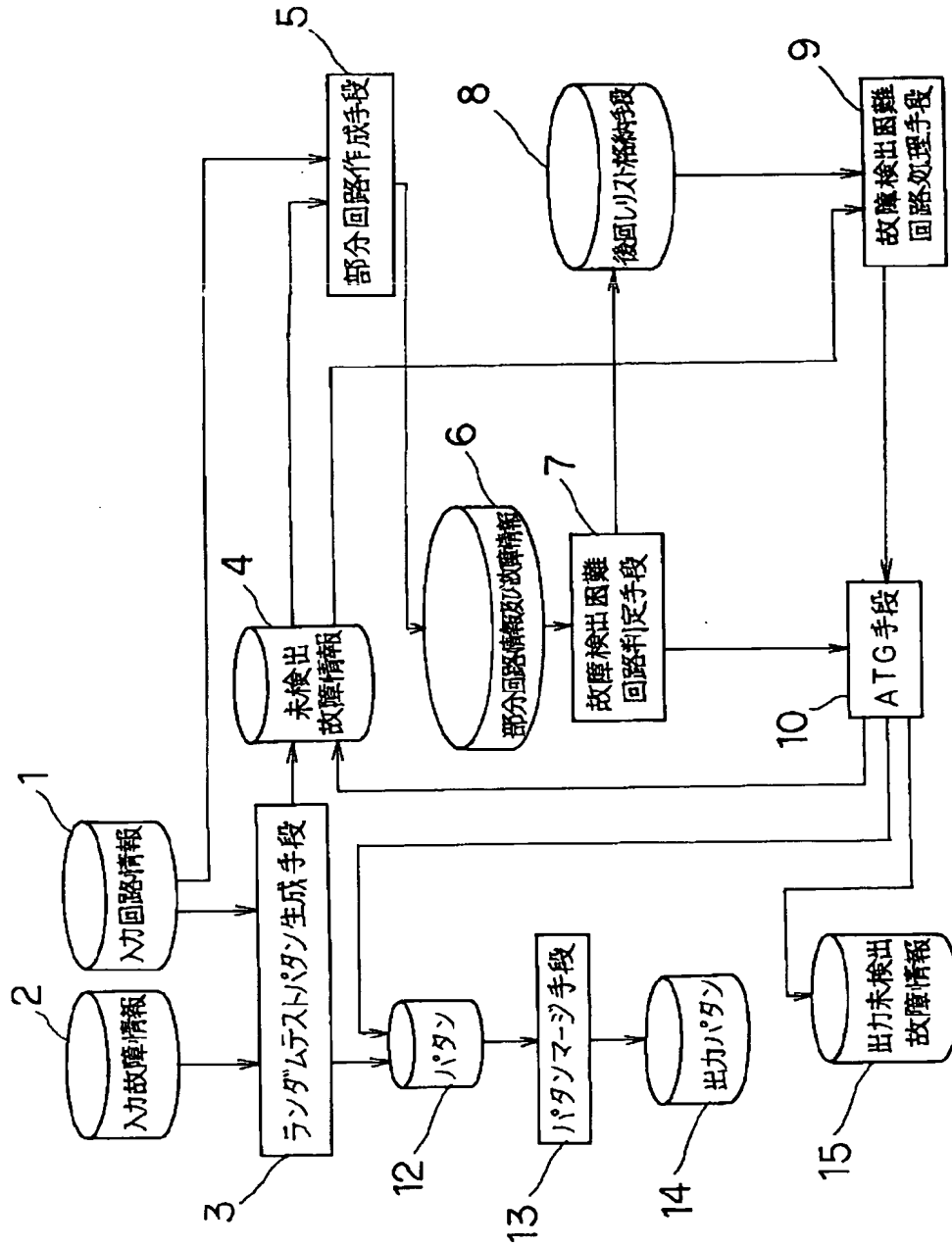
【符号の説明】

- 1 入力回路情報
- 2 入力故障情報
- 3 ランダムテストボタン生成手段
- 4 未検出故障情報
- 5 部分回路作成手段
- 6 部分回路情報及び故障情報
- 7 故障検出困難回路判定手段
- 8 後回しリスト格納手段
- 9 故障検出困難回路処理手段
- 10 ATG手段
- 12 ボタン
- 13 ボタンマージ手段

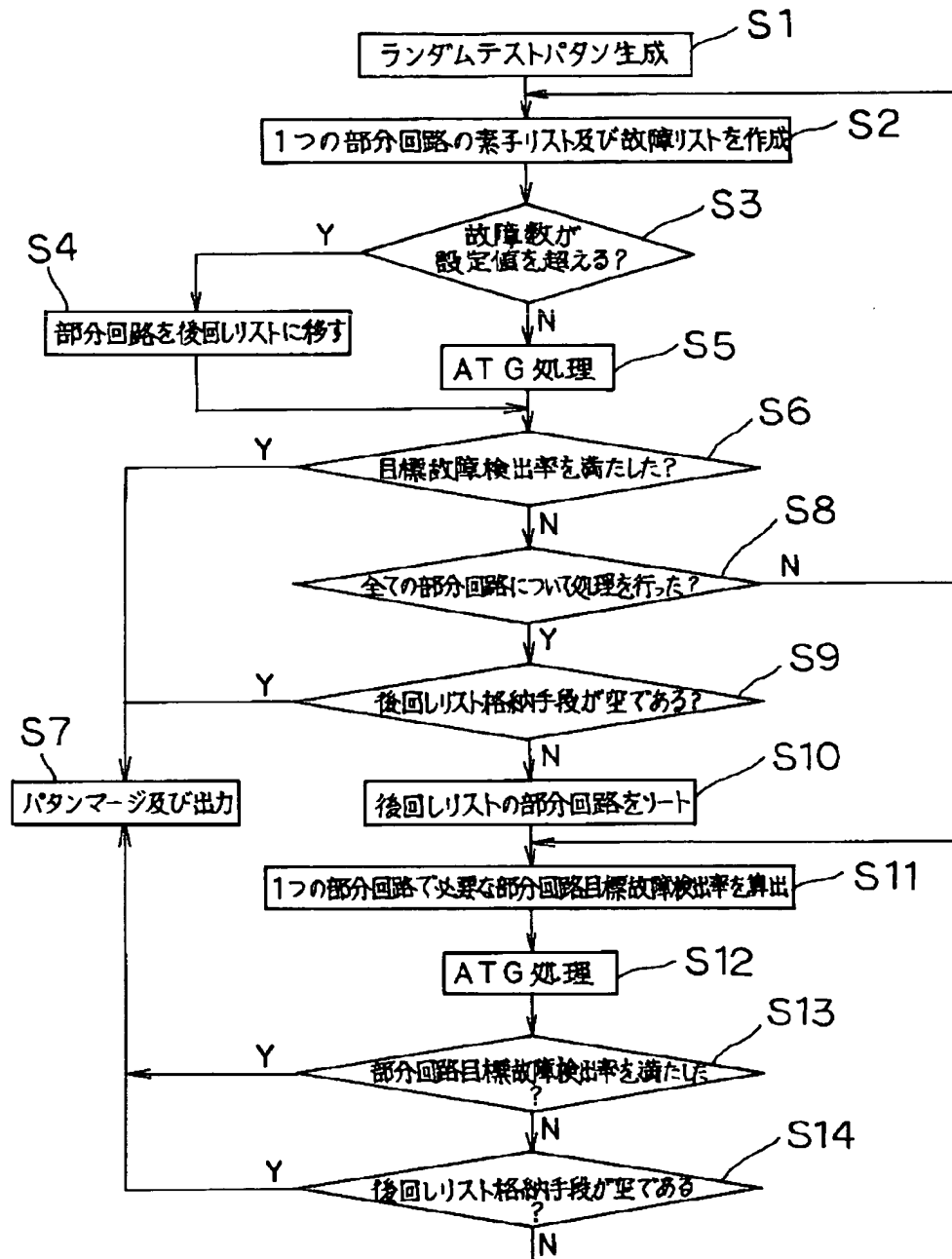
14 出力パターン

15 出力未検出故障情報

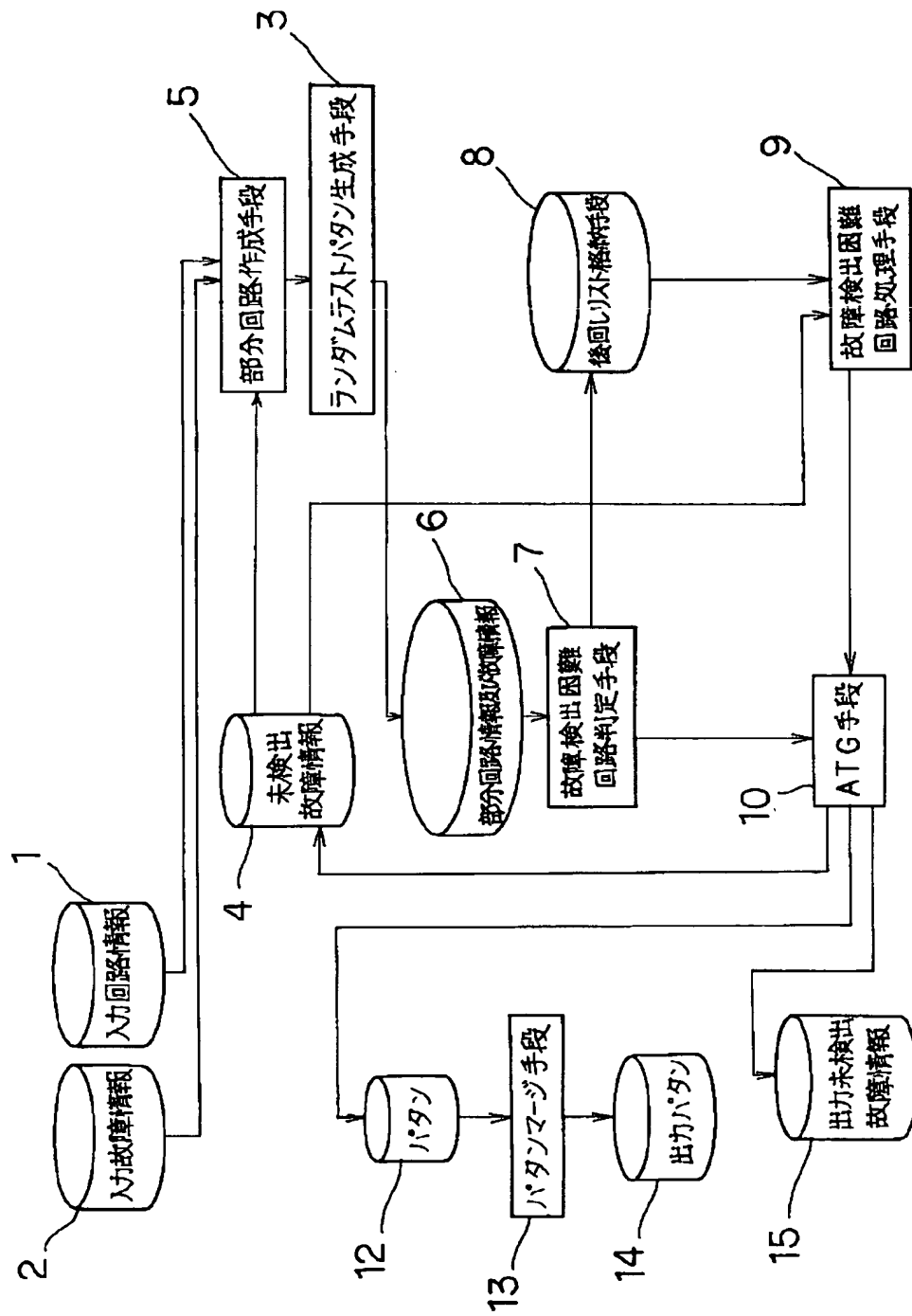
【図1】



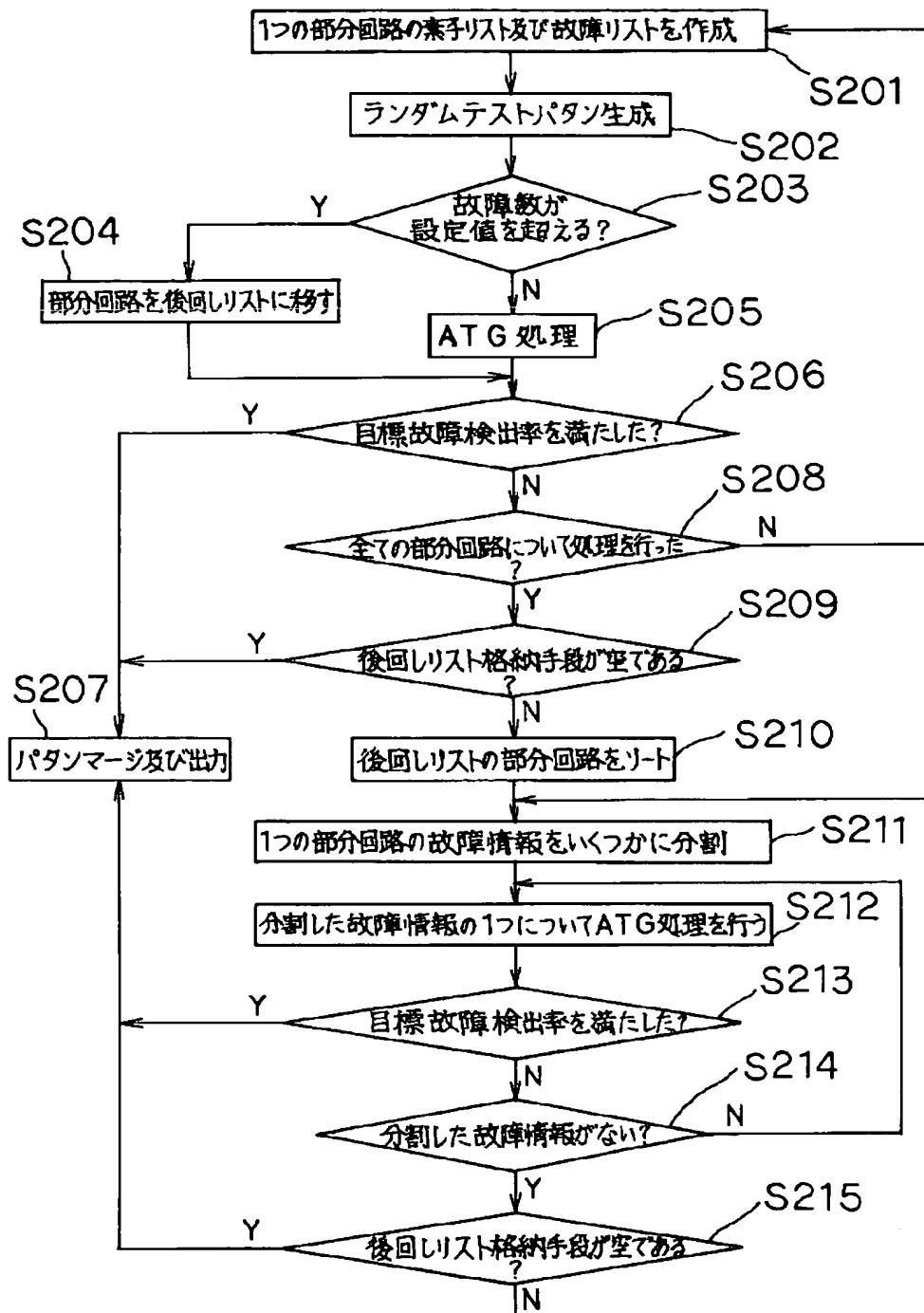
【図2】



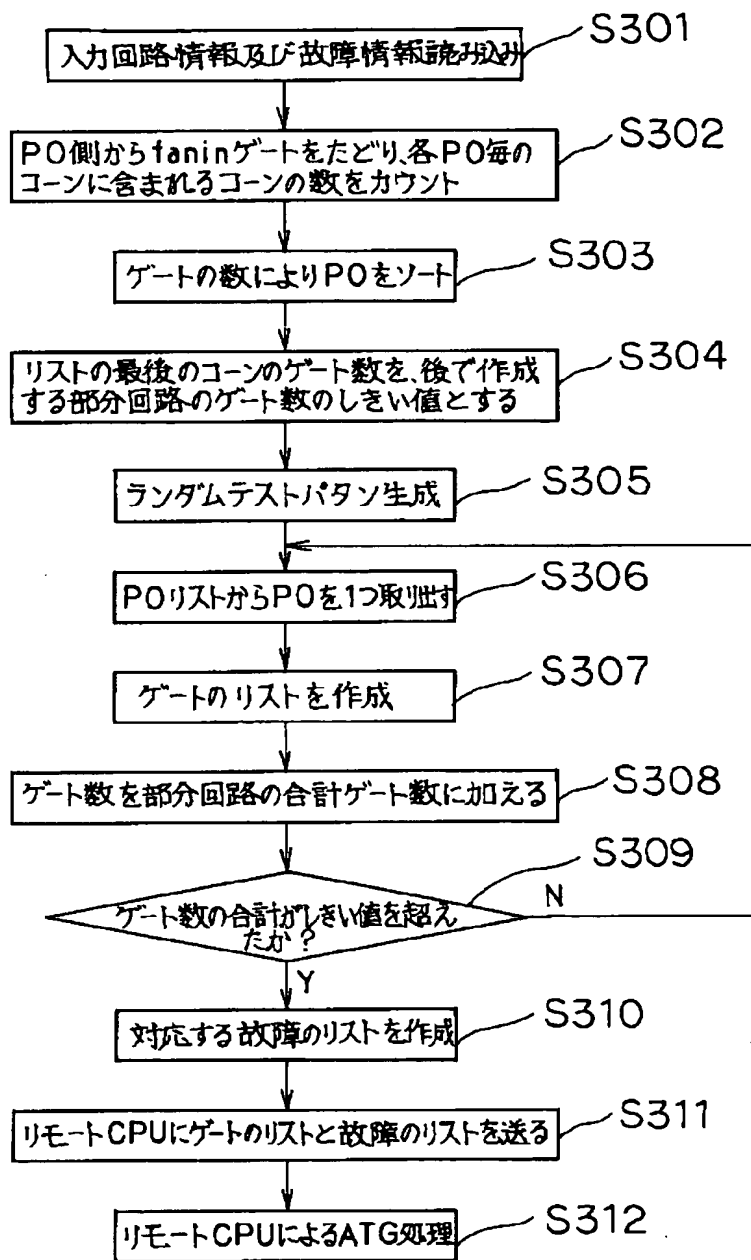
【図3】



【図4】

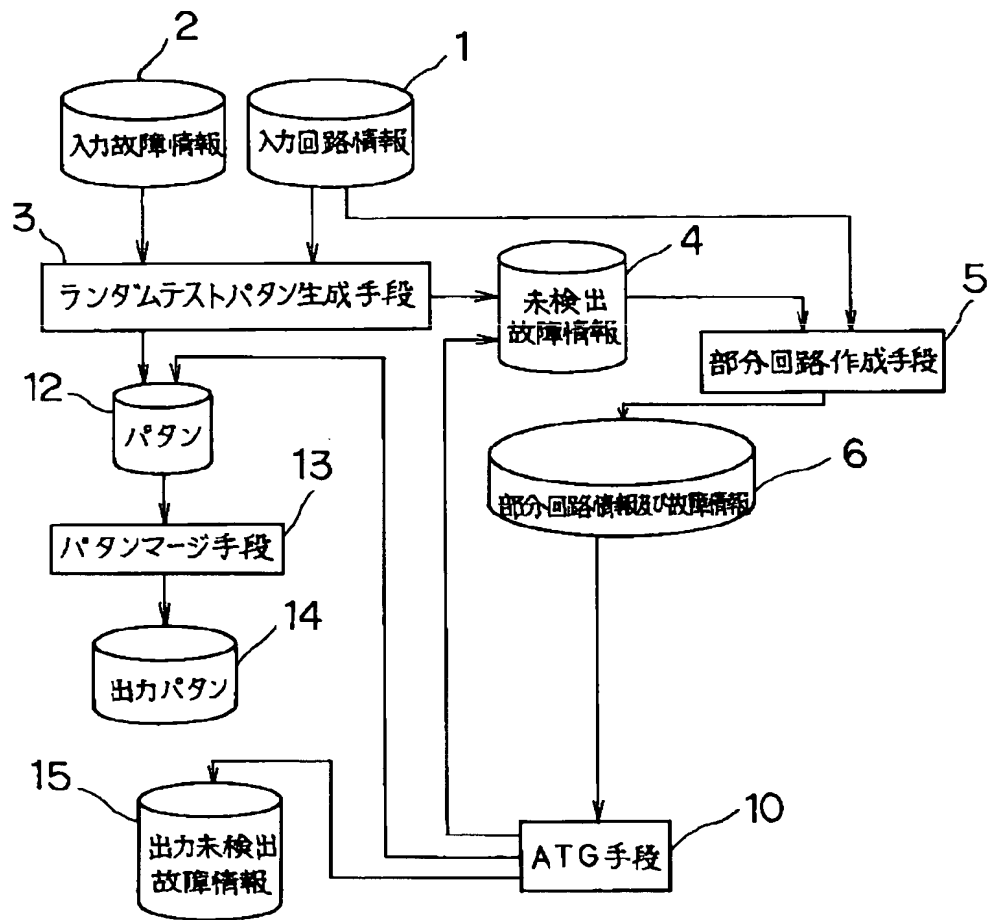


【図5】





【図6】



【図7】

